

BAB II

LANDASAN TEORI

2.1 Service Oriented Architecture (SOA)

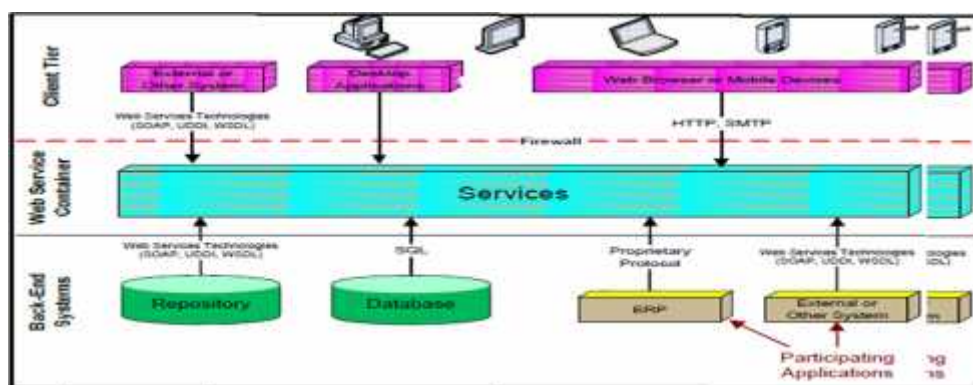
Menurut Thomas Erl (2005) Service Oriented Architecture (SOA) adalah bagian utama dari *service computing platform* yang membawa konsep, teknologi, dan tantangan baru baru. Menurut Thomas Erl ada tiga hal penting yang menjadikan sebuah infrastruktur dapat disebut sebagai *service oriented architecture*, yaitu logika bisnis yang dienkapsulasi sebagai *service*, dan proses komunikasi antar service dengan menggunakan *message*. Dalam hal ini, *service layer* akan menjembatani hubungan antara *business logic* dan *application logic*.

Menurut Kunal Mittal (2007), *Service Oriented Architecture* adalah sebuah kumpulan yang terdiri atas *tools*, teknologi, *framework*, dan *best practice* yang memudahkan implementasi sebuah *service* secara cepat. Proses dalam mengimplementasi SOA menggunakan metodologi yang mengidentifikasikan *service* yang dapat dipergunakan kembali (*reusable*) dalam aplikasi dan organisasi suatu perusahaan. Dengan demikian, SOA adalah suatu ide, bukan merupakan teknologi, produk, ataupun standar. Arsitektur SOA difokuskan untuk mengidentifikasi, membangun, mengubah, dan memelihara proses bisnis suatu perusahaan sebagai sekumpulan *service*. Teknologi yang menggunakan SOA digunakan untuk mengurangi kompleksitas dalam membangun sebuah aplikasi atau *software*. SOA dapat mengantisipasi isu mengenai penggunaan *software* yang terdistribusi, penggunaan *platform* yang berbeda, dan integrasi aplikasi.

Dapat disimpulkan bahwa SOA adalah suatu cara mengorganisir perangkat lunak (*software*) sehingga organisasi dapat dengan cepat merespon perubahan

kebutuhan. Teknologi tersebut berdasarkan *service* (layanan), yang terdiri dari unit-unit berdasarkan kebutuhan dari perangkat lunak yang berjalan pada jaringan. *Service* sendiri merupakan komponen umum yang digunakan oleh beberapa sistem aplikasi (*reusable*). *Service* dapat berupa modul program, aplikasi, atau gabungan dari beberapa aplikasi yang berhubungan. SOA merepresentasikan suatu model yang mana fungsi-fungsi dibagi menjadi beberapa unit-unit terpisah yang lebih kecil, yang dapat didistribusikan melalui jaringan dan dapat dikombinasikan dan digunakan secara bersama-sama untuk menciptakan aplikasi. *Service-service* tersebut berkomunikasi satu sama lain dengan cara mengirim data dari satu *service* ke *service* lainnya, atau dengan mengkoordinasikan suatu aktivitas antara dua atau lebih *service*. Sehingga SOA memungkinkan *service* yang *interoperable*, yang berarti *service-service* tersebut dapat berkomunikasi satu sama lain, meskipun pada implementasinya dibuat dengan bahasa pemrograman yang berbeda atau diakses melalui *transport protocol* yang berbeda yang memungkinkan pengintegrasian aset-aset sistem aplikasi dari suatu perusahaan

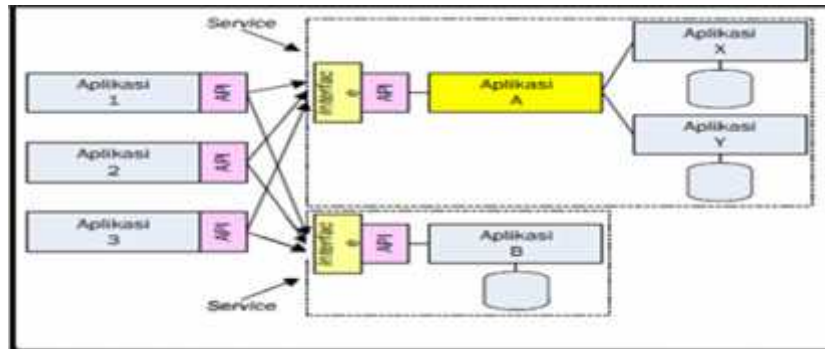
Ilustrasi konsep dari *Service Oriented Architecture* (SOA) dapat dilihat pada gambar dibawah ini :



Gambar 2.1 konsep *service oriented architecture* (sumberr: Thomas Erl, 2005)

Service merupakan komponen umum yang digunakan oleh beberapa sistem aplikasi (*reusable*). *Service* dapat berupa modul program, aplikasi, atau gabungan

dari beberapa aplikasi yang berhubungan. *Service* direalisasikan dengan menambahkan *interface (wrapper)* pada satu atau sekelompok sistem aplikasi. Sistem aplikasi berkomunikasi dengan *service interface* melalui API (*application programmer interface*).



Gambar 2.2 sistem aplikasi berkomunikasi dengan API (sumber: Thomas Erl, 2005)

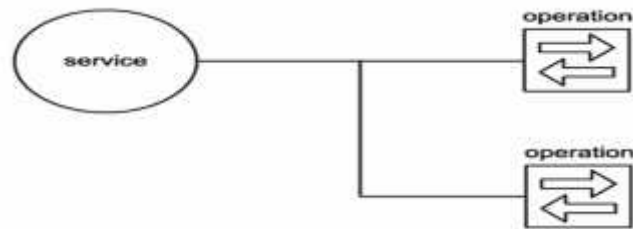
Service merupakan infrastruktur karena:

- Dipakai bersama oleh struktur di atasnya (sistem-sistem aplikasi).
- Relatif lebih stabil/statis dibandingkan aplikasi-aplikasi penggunaanya.
- Memiliki *life cycle* tersendiri.
- Dikelola (dirancang, dikembangkan, dan dipelihara) oleh pihak yang berbeda dengan pengelola aplikasi di atasnya.
- Dapat di-*outsource*.
- Menerapkan *Service Level Agreement*.

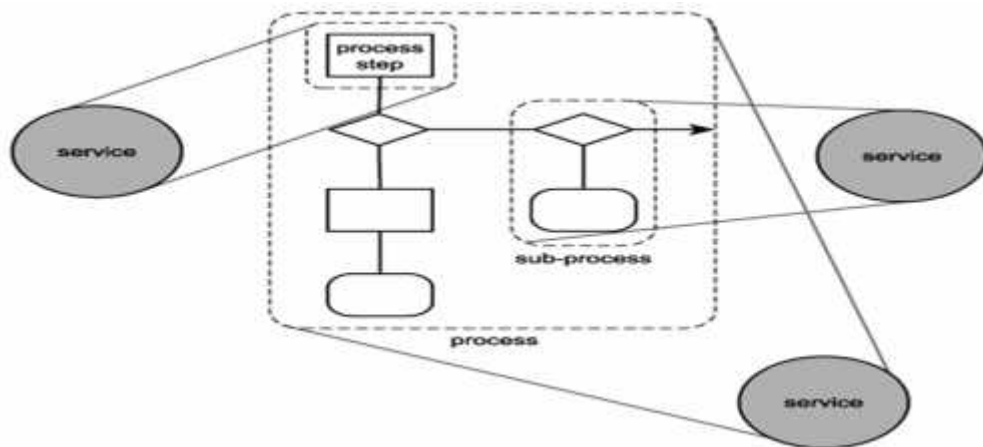
Dengan menyediakan *service*, keuntungan yang didapat antara lain:

- Dapat memanfaatkan ulang keahlian
- Meningkatkan kecepatan(*agility*) pengembangan sistem aplikasi
- Menurunkan kompleksitas pengembangan sistem aplikasi.

Untuk memahami SOA lebih lanjut, *webservice* akan dibahas pada bagian berikutnya.



Gambar: 2.3 *Service dan Operation* (sumber: Thomas Erl, 2005)



Gambar 2.4: *Service, Process, Process Step, Sub-process* (sumber: Thomas Erl, 2005)

2.1.1 Keuntungan menggunakan SOA

Keuntungan Menggunakan SOA

1. *Improved integration (and intrinsic interoperability)*

Dengan menggunakan SOA, *cost* dan *effort* dari integrasi *cross-application* akan menjadi menurun.

2. *Inherent reuse*

Arsitektur berorientasi layanan mempunyai sifat *services* yang dapat digunakan kembali (*reuse*).

3. *Streamlined architectures and solutions*

Implementasi SOA berpotensi untuk mengurangi *processingoverhead* dan mengurangi *skill-set* yang dibutuhkan (*technical resource* hanya membutuhkan pengetahuan pembuatan aplikasi, *service* atau *service extension*)

4. *Leveraging the legacy investment*

Cost dan *effort* untuk merespon dan beradaptasi ke bisnis atau *teknologi related change* berkurang.

5. *Establishing Standarized XML data representation*

Biaya dan usaha pengembangan aplikasi berkurang setelah pengembangan standar representasi dataXMLtercapai.

6. *Focus investment on communication infrastructure*

Biayauntuk memperbesar infrastruktur komunikasi berkurang, karena hanya satuteknologi komunikasi diperlukan untuk mendukung keseluruhan perusahaan.

7. *“Best-of-Bread” alternatives*

Peningkatan potensi pemenuhan kebutuhan bisnis, sejalan dengan peningkatan kualitas otomatisasi bisnis

8. *Organizational Agility*

Biaya dan usaha untuk merespon dan beradaptasi kepada perubahan bisnis atau teknologi berkurang.

2.1.2 Prinsip Dasar Layanan

1. *Services are reusable*

Service dapat memiliki satu atau banyak *operations*, tiap tiap *operations* dapat diakses oleh banyak *servicerequestors*

2. *Serviceshare formal contract*

Kontrak *service* menyediakan definisi formal dari:

- Endpoint dari *service*
- *Operations* yang terdapat pada tiap tiap *service*
- Setiap *input* dan *outputmessage* yang didukung oleh tiap tiap *operations*
- Aturan dan karakteristik dari *service* dan *operation-operation* yang terdapat di dalamnya

3. *Services are loosely coupled*

Loose coupling adalah kondisi dimana sebuah *service* mendapatkan informasi dari *service* lainnya dalam kondisi independen dari *service* tersebut. Kondisi *loose coupling* dapat diperoleh dari penggunaan kontrak *service* yang memungkinkan *services* berinteraksi tanpa mendefinisikan parameter-paramenter yang diperlukan terlebih dahulu.

4. *Services abstract underlying logic*

Service tidak memperlihatkan bagaimana *logic* diimplementasi di dalamnya

5. *Services are composable*

Konsep yang mendukung *composability* adalah konsep orkestrasi, yaitu proses *service-oriented* yang dikontrol oleh sebuah induk proses yang terdiri dari beberapa anggota *services*

6. *Serviceare autonomous*

Service memiliki hak penuh terhadap semua *logic* yang dienkapsulasi

7. *Serviceare stateles*

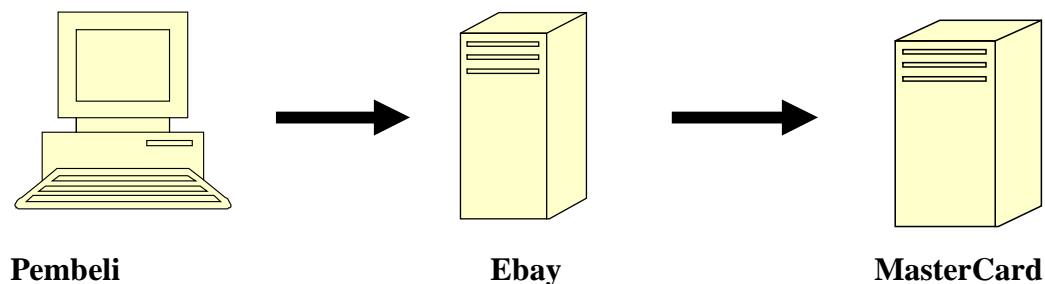
Service tidak memiliki status tertentu terkait dengan aktifitas yang dilakukan

Web Service

Sesuai dengan namanya, *web service* hanya menyediakan *service* atau layanan. *Web service* dibuat bukan untuk berinteraksi langsung kepada *client*, tetapi untuk menyediakan layanan yang akan dipanggil oleh aplikasi. *Web service* dapat didefinisikan sebagai aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet dengan menggunakan XML sebagai format pengiriman pesan

Pada kondisi dimana operating system dan bahasa pemrograman beraneka ragam jenisnya, timbul masalah dimana pertukaran data antar perangkat yang menggunakan aplikasi dan platform yang berbeda. Inilah alasan utama menggunakan *web service*, *web service* memungkinkan penggunaan *platform* yang berbeda dapat bertukar data dan informasi dengan mudah.

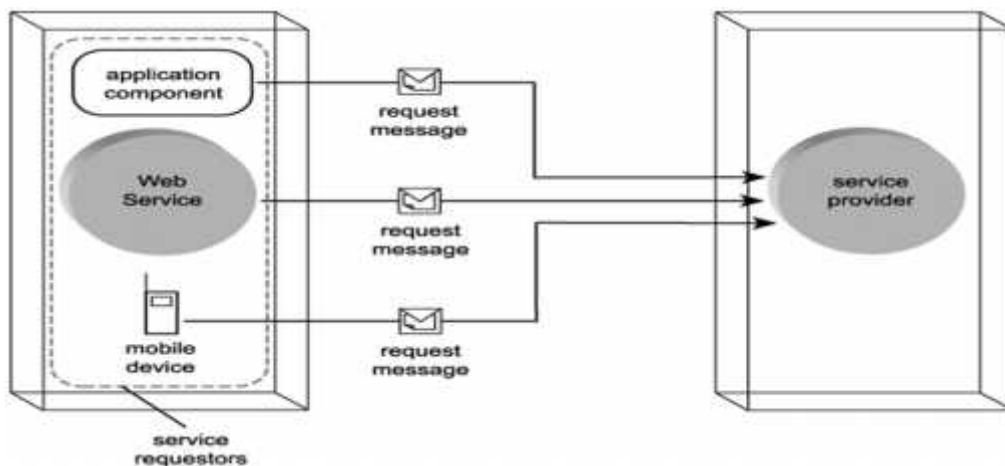
Contoh aplikasi dari *web service* adalah Ebay.com. Pada waktu pembeli ingin melakukan proses pembayaran, Ebay meminta nomor kartu kredit yang dimiliki oleh pembeli. Ebay akan mengecek masa berlaku kartu kredit pembeli. Disinilah peran *web service* sebagai penghubung proses transaksi antar penerbit kartu kredit tersebut (sebut saja MasterCard). MasterCard akan menyediakan sebuah *webservice* untuk melakukan pengecekan masa berlaku kartu kredit.



Gambar: 2.5 contoh aplikasi dari web service

Pada prinsipnya peranan *service* dibagi menjadi dua, yaitu *serviceprovider* dan *servicerequestor*. *Serviceprovider* adalah *service* yang menyediakan layanan

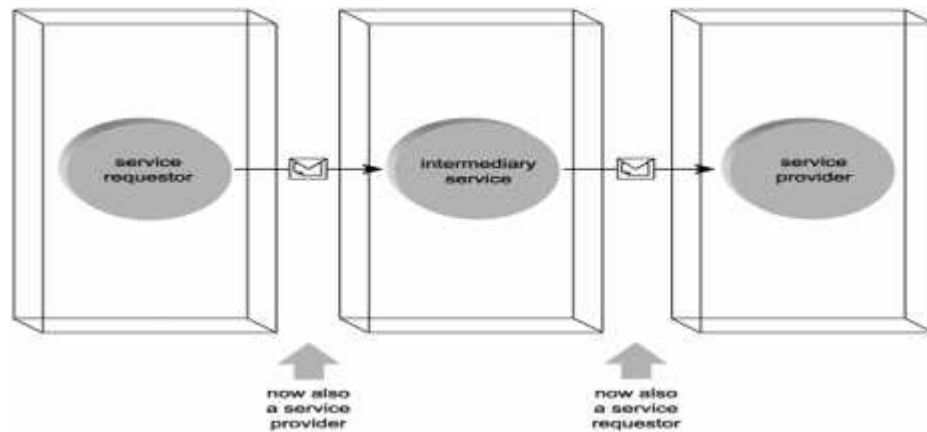
yang akan digunakan dan *servicerequestor* adalah *service* yang meminta *service* tersebut berjalan. Alokasi *provider* dan *requestor* tidak bersifat mutlak. Sebuah *serviceprovider* dapat berubah menjadi *servicerequestor*, tergantung dari peran yang diberikan oleh proses bisnis yang terkait.



Gambar 2.6: Alur *request* dari *servicerequestor* ke *serviceprovider*
(sumber: Thomas Erl, 2005)

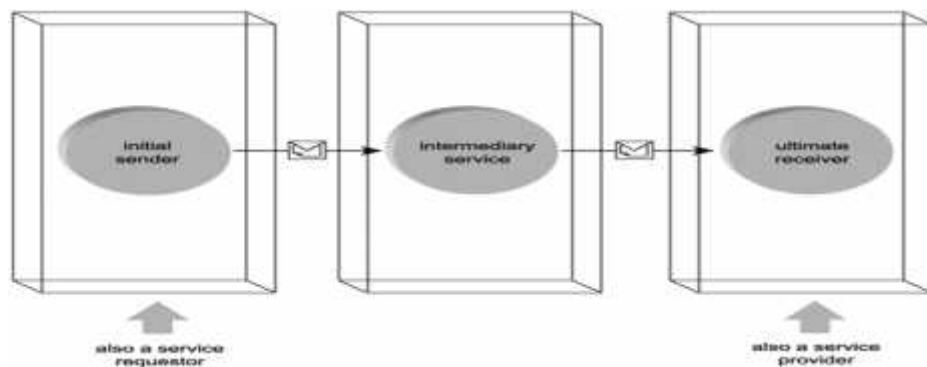
Requestor dari sebuah *service* tidak hanya berasal dari sebuah *service*, tetapi juga dapat berasal dari aplikasi *native* atau *mobile*. Pada Gambar diatas dijelaskan bahwa *servicerequestor* meminta *service* kepada *serviceprovider* dengan menggunakan *request message* berbentuk amplop. Amplop tersebut adalah penggambaran dari SOAP yang berupa struktur bahasa dari XML yang digunakan untuk berkomunikasi oleh *servicerequestor* dan *serviceprovider*. Gambar adalah sebuah implementasi sederhana dari sebuah *servicerequestor* kepada *serviceprovider*.

Peran *intermediaries* (penghubung) akan sering dijumpai seiring semakin kompleks-nya sebuah sistem. Semakin kompleks sebuah sistem, maka akan semakin banyak *services* yang digunakan dan hubungan antar *service* akan semakin fleksibel.



Gambar 2.7: *Intermediaryservice* (sumber: Thomas Erl, 2005)

Pada Gambar dijelaskan peranan *intermediaryservice* pada sesi pertama dimana *servicerequestor* meminta *service* pada *intermediaryservice* dimana *intermediaryservice* bertindak sebagai *serviceprovider*. Setelah request diterima *intermediaryservice* meneruskan permintaan kepada *serviceprovider*, pada tahap kedua *intermediaryservice* bertindak sebagai *servicerequestor*.

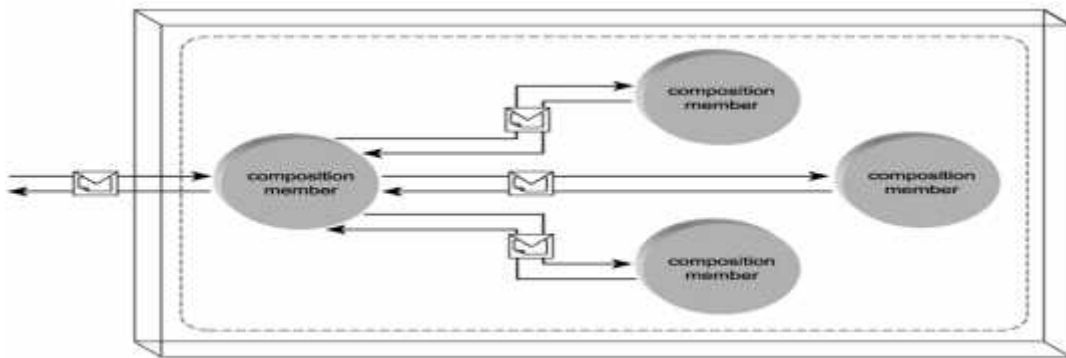


Gambar 2.8: *Initial sender, intermediaryservice* dan *ultimate receiver*
(sumber: Thomas Erl, 2005)

Terkadang peran antar *services* dideskripsikan sebagai *intial sender*, *intermediaryservice* dan *ultimate receiver*. *Initial sender* yaitu *servicerequestor* yang pertama kali mengirimkan permintaan *service* kepada *intermediaryservice*. Setelah permintaan diterima, *intermediaryservice* akan bertindak sebagai *servicerequestor*,

tapi tidak sebagai *initial sender*. *Ultimate receiver* akan menerima request yang diteruskan oleh *intermediaryservice*. *Intermediaryservice* tidak akan menjadi *initial sender* atau *ultimate sender* dalam area aktifitas *service*

Pada contoh Gambar, 2.8, Gambar diperlihatkan hubungan antara satu *service* dengan *service* lainnya. Selain hubungan satu-satu, *services* dapat berhubungan secara kolektif, bersama-sama bekerja untuk menyelesaikan suatu pekerjaan.



Gambar 2.9 *service* secara kolektif (sumberr: Thomas Erl, 2005)

Pada Gambar 2.9 dijelaskan banyak *service* bekerja untuk menyelesaikan suatu pekerjaan, sebuah *service* yang menerima request meneruskan *request* tersebut kepada tiga *services* lainnya. Ketiga *services* tersebut akan mengirimkan responses yang diminta sebelumnya dan dikumpulkan disebuah *service*, yang nantinya akan diteruskan kepada *serviceprovider* sebagai *serviceresponse*.

Kelebihan dari web *service* adalah:

1. Lintas platform

Dengan menggunakan web *service*, komputer yang memiliki sistem operasi yang berbeda dapat saling berkomunikasi.

2. Language independent

Akses kepada web *service* tidak terbatas pada hanya satu bahasa pemrograman saja. Sebagai contoh, web *service* yang dibangun dengan PHP dapat diakses oleh PHP, JSP, Delphi dan VB.NET.

3. Jembatan penghubung dengan *database*

Webservice dapat menjadi penghubung antara aplikasi dengan *database*. Dengan menggunakan *webservice*, aplikasi tidak memerlukan driver *database* dan tidak perlu mengetahui *database* yang digunakan oleh *server* serta bagaimana struktur *database* tersebut jika ingin mengaksesnya.

4. Mempermudah proses pertukaran data

Pertukaran data antara dua perusahaan yang memiliki platform yang berbeda dapat terjadi dengan menggunakan *webservice*.

5. Penggunaan kembali komponen aplikasi

Sebuah layanan dapat melayani sebuah fungsi untuk beberapa pengguna layanan. Sebagai contoh, layanan konversi mata uang dapat digunakan oleh bank A dan bank B.

Beberapa komponen pendukung *web service* adalah:

1. XML

Extensible Markup Language (XML) diciptakan pada akhir tahun 60-an oleh W3C. XML menjadi populer pada *ebusiness* pada akhir 90-an, pada saat transaksi online menggunakan bahasa *server-side scripting* menjadi tren.

Arsitektur data yang direpresentasikan XML mencerminkan pemahaman dasar terhadap SOA. Dalam arsitektur data tersebut XML membangun format dan struktur dari *message* yang berjalan antar *service*.

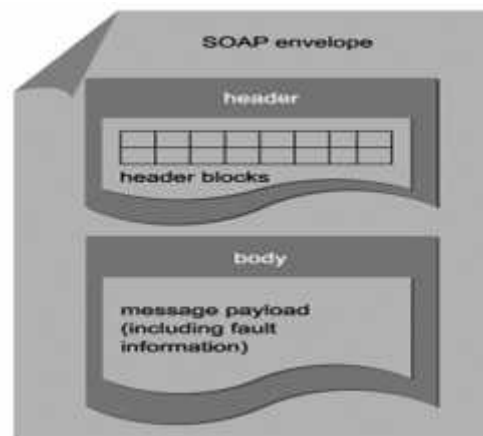
XML bersifat *platform independent*, informasi pada XML dapat diakses oleh platform apapun selama aplikasi tersebut dapat menterjemahkan *tag* XML. Dalam *web service* XML dapat diumpamakan sebagai bahasa Inggris yang menjadi bahasa internasional, dimana orang diseluruh dunia mengerti akan bahasa tersebut.

2. SOAP

SOAP (*Simple Object Access Protocol*) merupakan format standar dokumen yang berbentuk XML yang digunakan untuk melakukan proses *request* dan *response* antar *webservice* dengan aplikasi yang memanggilnya (Lucky, 2008). Komunikasi antar *services* yang menggunakan XML harus distandarisasi, dengan menggunakan SOAP, komunikasi antar *services* akan lebih teratur. Jika diibaratkan XML adalah bahasa inggris, maka SOAP adalah *grammar* atau struktur bahasa.

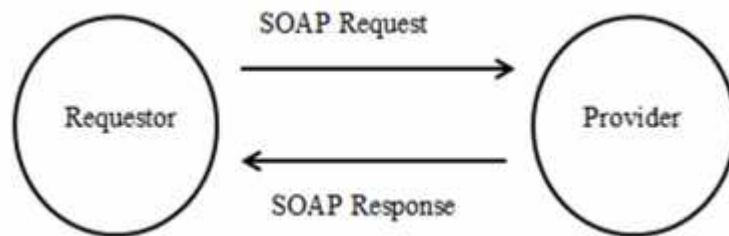
Tujuan utama penggunaan SOAP adalah menstandarisasi format *message* yang berjalan antar *service*. Struktur dari SOAP terdiri dari SOAP *envelope*, *header* dan *body*.

Secara arti kata, *envelope* berarti amplop, yaitu dimana disimpan sebuah surat, pada SOAP *envelope* menyimpan seluruh informasi *message* yang mengalir. Informasi penting disimpan pada *body*, dimana terdiri dari data berformat XML. Dan *header* dimana informasi yang bersifat tambahan disimpan.



Gambar 2.10 SOAP (sumberr: Thomas Erl, 2005)

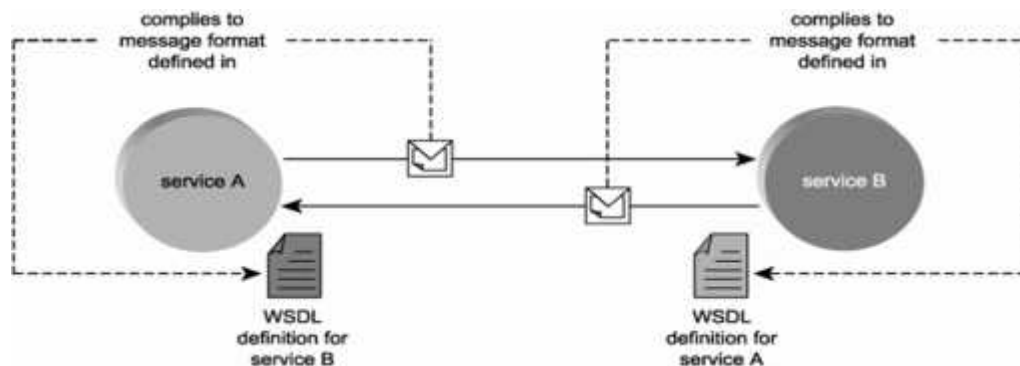
Dokumen SOAP yang digunakan untuk melakukan sebuah permintaan (*request*) oleh *servicerequestor* disebut *SOAP request* dan dokumen SOAP yang diterima dari *serviceprovider* disebut SOAP response.



Gambar 2.11 SOAP *request* dan *response* (sumberr: Thomas Erl, 2005)

3. WSDL

Web Service Description Language (WSDL) adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah *webservice*. Didalam WSDL dijelaskan *method-method* apa saja yang tersedia didalam *web service*, parameter apa saja yang diperlukan untuk memanggil sebuah *method*, dan apa hasil atau tipe data yang dikembalikan oleh *method* yang dipanggil tersebut. (Lucky, 2008)



Gambar 2.12 WSDL (sumberr: Thomas Erl, 2005)

Pada Gambar dijelaskan hubungan antara *service A* dan *service B*. pada SOAP request yang berasal dari *service A* terdapat *WSDL definition* untuk *service B*.

dan pada SOAP response yang berasal dari *service* B terdapat WSDL *definition* untuk *service* A.

Dalam pengertian sederhana, WSDL dapat berupa sebuah brosur dan formulir sebuah perusahaan. Brosur menjelaskan layanan apa saja yang dapat diberikan oleh perusahaan tersebut dan keuntungan memakai layanan dari perusahaan. Dan formulir merupakan informasi yang harus diisi oleh pengguna layanan yang akan digunakan oleh perusahaan demi memberikan pelayanan yang maksimal.

2.3 Web Iklan

Web iklan adalah sebuah web yang hanya memperkenalkan (produk) barang, layanan, kesempatan kerja, inspirasi dan informasi kepada yang ingin disampaikan kepada khalayak masyarakat. Jadi lebih sederhananya adalah makelar, artinya web iklan hanya memertemukan antara penjual dengan pembeli melalui media online. Sehingga yang berjualan pada situs jual beli adalah masyarakat luas, bukan pemilik situs jual beli. Web Iklan memfasilitasi masyarakat yang ingin menjual barangnya secara online, dengan cara memberikan tempat bagi masyarakat luas untuk mempromosikan dan memasarkan barang yang hendak dijual secara online. Kegiatan jual beli seperti tawar menawar harga, cara pembayaran, cara pengiriman, murni ada di tangan pemasang iklan dan calon pembeli. Maka dari itu pembayaran dari barang yang akan dibeli pun mengarah kepada penjual barang, bukan ke pemilik situs jual beli.

2.3.1 Web Provider/Serbaiklan.Com

Web provider/serbaiklan.com merupakan web yang bergerak di bidang periklanan yang berfungsi sebagai media promosi. Dan memiliki sebuah layanan yang bisa terintegrasi dengan web client. Fasilitas yang ada pada web serbaiklan ini yaitu sebuah aplikasi web service yang menunjang penyebaran informasi ke dalam web client. Aplikasi web service ini hanya menyediakan fasilitas kepada data buku. Yang nantiya setiap perubahan data buku akan langsung tersebar ke web client.

Berikut ini contoh-contoh web yang berkaitan tentang media promosi online:

a. Toko Bagus



Gambar 2.13 Web Iklan Tokobagus

Tokobagus.com adalah pusat jual beli online terbesar di Indonesia. Tempat untuk mencari barang baru dan bekas berkualitas seperti produk handphone murah, komputer, fashion, mobil bekas, motor, rumah dan properti, peralatan rumah tangga, aneka jasa dan juga lowongan kerja. Untuk para penjual, pasang iklan gratis adalah salah satu layanan yang disediakan oleh Tokobagus. Iklan anda akan dilihat oleh ratusan ribu orang setiap harinya. Bertransaksi di Tokobagus, baik jual maupun beli tidak dikenakan biaya, semua disediakan gratis. Bergabunglah sekarang juga bersama jutaan member lain yang sudah terdaftar.

b. Berniaga



Gambar 2.14 Web Iklan Berniaga

Berniaga.com mempertemukan penjual dan pembeli agar mendapatkan penawaran terbaik dengan cara yang mudah dan aman. Begitu mudahnya memasang iklan dan mencari barang yang diperlukan di situs berniaga darimanapun dan kapanpun. Anda dapat berhubungan langsung dengan penjual/pembeli dan mendapat penawaran terbaik. Berniaga juga berusaha menjaga dan melindungi seluruh pengunjung situs agar terhindar dari penipuan. Fasilitas keamanan yang disiapkan adalah dengan mereview setiap iklan yang masuk agar iklan yang dipublikasikan nantinya benar-benar berkualitas. Hal ini meminimalisir potensi penipuan yang kerap terjadi pada jual beli online.

c. Kaskus



Gambar 2.15 Web Iklan Kaskus

Kaskus adalah forum diskusi dan jual beli terbesar di Indonesia. Apapun tujuan anda ke Kaskus, maka yang diperlukan pertama kali adalah daftar membuat akun Kaskus. Setelah akun Kaskus aktif, maka anda dapat berdiskusi, jual beli dan pasang iklan di forum tersebut. Menurut Alexa, situs Kaskus menempati ranking tertinggi mengalahkan Detik.com yang berada di tempat kedua.

2.4 T4jualan.com/web client

Web t4jualan.com adalah merupakan salah satu situs Online Shop dimana perusahaan dan perorangan dapat menjual dan membeli produk maupun jasa yang memberikan keleluasaan bagi para penggunanya. t4jualan.com mencoba secara terus menerus dalam memberikan jasa pelayanan yang terbaik untuk Anda para user dalam saling bertransaksi.

Visi:

Menjadi jaringan distribusi retail terkemuka yang dimiliki oleh masyarakat luas, berorientasi kepada pemberdayaan pengusaha kecil, pemenuhan kebutuhan dan harapan konsumen, serta mampu bersaing secara global.

Misi:

1. Memberikan kepuasan kepada pelanggan atau konsumen dengan berfokus pada produk dan pelayanan yang berkualitas unggul.
2. Selalu menjadi yang terbaik dalam segala hal yang dilakukan dan selalu menegakkan tingkah laku / etika bisnis yang tertinggi.
3. Ikut berpartisipasi dalam pembangunan negara dengan menumbuhkan-kembangkan jiwa wiraswasta dan kemitraan usaha.
4. Membangun organisasi global yang terpercaya, tersehat dan terus bertumbuh dan bermanfaat bagi pelanggan, pemasok, karyawan, dan masyarakat pada umumnya.



Gambar 2.16 Web Penjualan T4jualan

Tampilan menu utama pada t4jualan.com berisikan tentang informasi-informasi tentang iklan. Dan menampilkan menu-menu yang akan dibutuhkan oleh pengguna web.



Gambar 2.17 Tampilan Menu Login Web Client/T4jualan.Com

Menu login berfungsi sebagai verifikasi pengguna untuk masuk kedalam web t4jualan.com. setelah login pengguna dapat mengakses seluruh menu yang ada di dalam web t4jualan.com



Gambar 2.18 Tampilan Form Daftar Iklan Web Client/T4jualan.Com

Form daftar iklan berfungsi untuk pengguna yang ingin mendaftarkan produk yang akan dipromosikan ke dalam web t4jualan.com

Berikut ini contoh-contoh web yang berikatan dengan web penjualan:

a. Gramedia



Gambar 2.19 Web Penjualan Gramedia Online

Toko Buku Gramedia Online merupakan bagian dari toko buku Gramedia yang tidak terpisahkan, menjadi satu kesatuan dalam menjalankan visi misi perusahaan (berperan aktif di dalam upaya mencerahkan kehidupan bangsa dengan menjadi jaringan ritel terbesar, tersebar dan terpadu di bidang pengetahuan, informasi dan multimedia). Kami menjual produk buku lokal maupun impor dari berbagai macam penerbit, produk stationary, alat olahraga, alat musik dan multimedia.

b. Amazon



Gambar 2.20 Web Penjualan Amazon

Amazon.com , Inc (Amazon.com) , didirikan pada 28 Mei 1996 , melayani konsumen melalui situs ritel dan fokus pada pemilihan , harga, dan kenyamanan . Perusahaan juga memproduksi dan menjual perangkat Kindle . Ia menawarkan program yang memungkinkan penjual untuk menjual produk mereka di website dan situs web bermerek mereka sendiri dan untuk memenuhi pesanan melalui mereka , dan program yang memungkinkan penulis , musisi , pembuat film , pengembang aplikasi , dan lain-lain untuk menerbitkan dan menjual konten .. Perusahaan beroperasi dalam dua segmen : Amerika Utara dan Internasional . Perusahaan menyajikan konsumen melalui situs ritel , dan fokus pada pemilihan , harga, dan kenyamanan .

2.5 Unified Modelling Language (UML)

Menurut Adi Nugroho (2005) secara umum UML merupakan “bahasa” untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. Dalam kerangka visualisasi, para pengembang menggunakan UML sebagai suatu cara calon pengguna sistem/perangkat lunak. Dengan adanya bahasa yang bersifat standar, komunikasi perancang dengan pemrogram serta calon pengguna diharapkan akan menjadi mulus.

2.5.1 Diagram UML

Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams. Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagram tersebut. Untuk menguasai UML, terdapat dua hal yang harus perhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

Tulisan ini pada intinya akan mengupas kedua hal tersebut. Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram- diagram sebagai berikut:

1. *Use Case Diagram*
2. *Class Diagram*
3. *Activity Diagram*
4. *Sequence Diagram*
5. *Collaboration Diagram*

2.5.1.1 Use Case Diagram

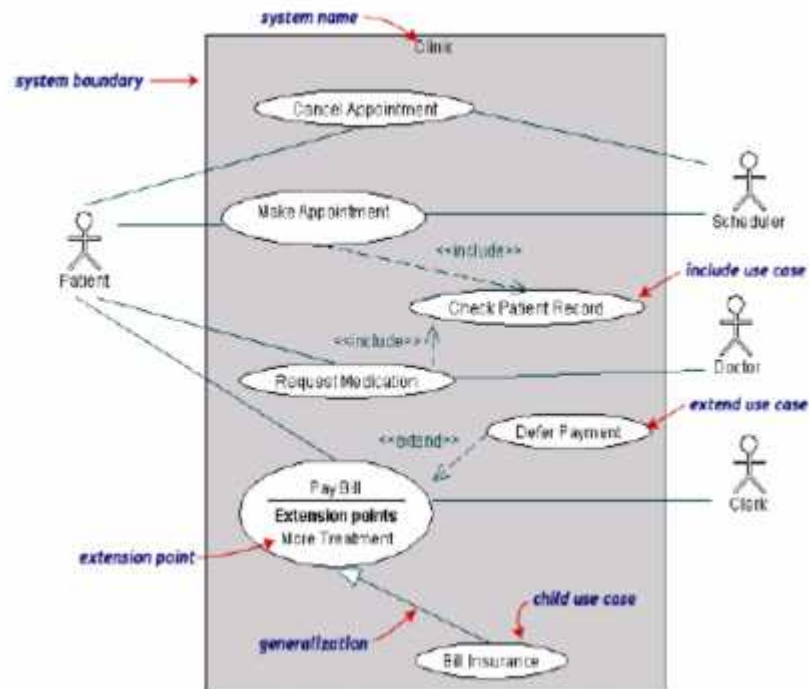
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan

“bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.



Gambar 2.21 Contoh *Use Case Diagram* (sumber : adi nugroho,2005)

2.5.1.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

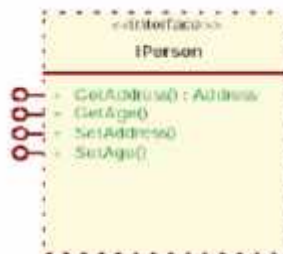
1. Nama (dan stereotype)
2. Atribut
3. Metoda, Atribut dan metoda dapat memiliki salah satu sifat berikut :
 - *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan

- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja



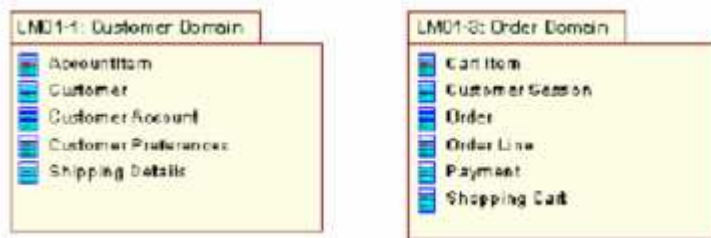
Gambar 2.22 Contoh *Class Diagram* (sumber : Adi nugroho,2005)

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.



Gambar 2.23 Contoh Implementasi *Class Diagram*
(sumber : adi nugroho,2005)

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

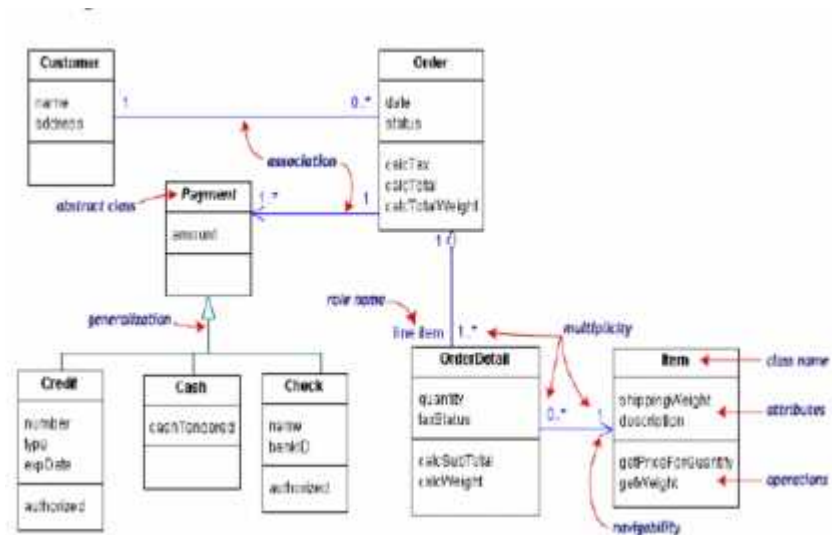


Gambar 2.24 Contoh *Class Diagram* Terdiri Atas *Package*
(sumber : adi nugroho,2005)

2.5.1.3 Hubungan Antar Class

Hubungan antar class dapat dikelompokkan menjadi 4 buah sebagaimana berikut ini:

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.



Gambar 2.25 Contoh Hubungan Antar Class (sumber : adi nugroho,2005)

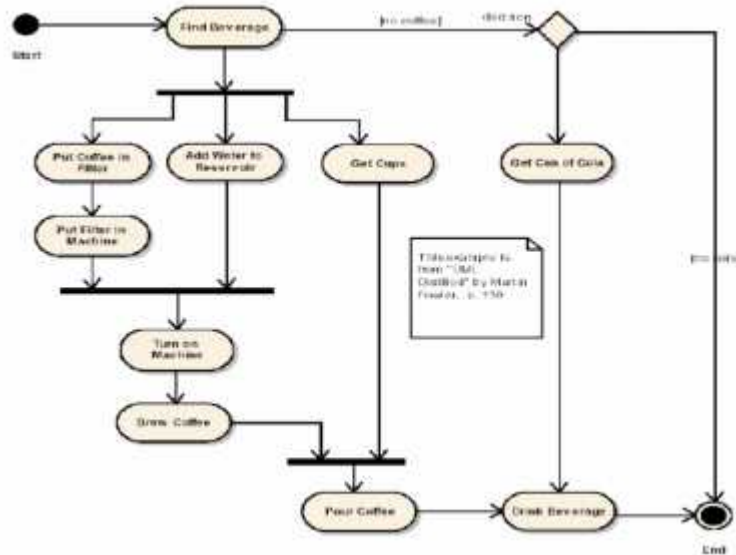
2.5.1.4 Activity Diagrams

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour

pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



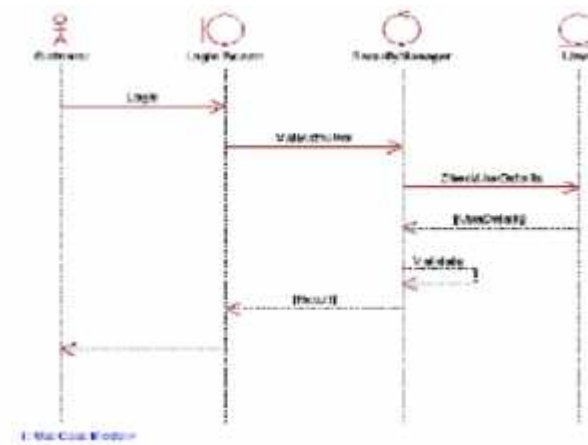
Gambar 2.26 Contoh *Activity diagrams* (sumber : adi nugroho,2005)

2.5.1.5 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 2.27 Contoh *Sequence diagram* (sumber : adi nugroho,2005)

2.5.1.6 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.



Gambar 2.28 Contoh *Collaboration Diagram* (sumber : adi nugroho,2005)

2.5.2 Langkah-Langkah Penggunaan Uml

Berikut ini adalah tips pengembangan piranti lunak dengan menggunakan UML:

1. Buatlah daftar *business process* dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk tiap *business process* untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian perhalus use case diagram dan lengkapi dengan *requirement*, *constraints* dan catatan-catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain (non-fungsional, *security* dan sebagainya) yang juga harus disediakan oleh sistem.
5. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.
6. Definisikan objek-objek level atas (*package* atau *domain*) dan buatlah *sequence* dan/atau *collaboration diagram* untuk tiap alir pekerjaan. Jika

sebuah *use case* memiliki kemungkinan alir normal dan error, buatlah satu diagram untuk masing-masing alir.

7. Buarlah rancangan *user interface* model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi hirarki *class* lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap *class* dibuat *unit test* untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
9. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokan *class* menjadi komponen-komponen. Karena itu buatlah component diagram pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia berinteraksi dengan baik. Perhalus *deployment diagram* yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam node.
10. Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan :
 - a. Pendekatan *use case*, dengan meng-*assign* setiap *use case* kepada tim pengembang tertentu untuk mengembangkan *unit code* yang lengkap dengan tes.
 - b. Pendekatan komponen, yaitu meng-*assign* setiap komponen kepada tim pengembang tertentu.
11. Lakukan uji modul dan uji integrasi serta perbaiki model berserta *codenya*. Model harus selalu sesuai dengan *code* yang aktual.
12. Piranti lunak siap dirilis.

2.5.1.8 Tools Yang Mendukung Uml

Saat ini banyak sekali tool pendesainan yang mendukung UML, baik itu *tool* komersial maupun *opensource*. Beberapa diantaranya adalah:

1. *Rational Rose* (www.rational.com)
2. *Together* (www.togethersoft.com)
3. *Object Domain* (www.objectdomain.com)
4. *Jvision* (www.object-insight.com)
5. *Objectteering* (www.objectteering.com)
6. *MagicDraw* (www.nomagic.com/magicdrawuml)
7. *Visual Object Modeller* (www.visualobject.com)

2.6 Internet (Interconnected Network)

Internet adalah sebuah jaringan besar yang terdiri dari berbagai jaringan yang meliputi jaringan bersifat bisnis, pendidikan dan riset serta menghubungkan jutaan komputer didalam jaringan jaringan tersebut (O'Brien,1997).

Setiap komputer dan jaringan terhubung secara langsung maupun tidak langsung ke beberapa jalur utama yang disebut internet backbone dan dibedakan satu dengan yang lainnya menggunakan unique name yang biasa disebut dengan alamat IP 32 bit. Contoh: 202.155.4.230.

Komputer dan jaringan dengan berbagai platform yang mempunyai perbedaan dan ciri khas masing-masing(Unix, Linux, Windows, Mac, dll) bertukar informasi dengan sebuah prtokol standar yang dikenal dengan nama TCP/IP(Transmissipon Control Protokol/ Internet Protokol). TCP/IP tersusun atas 4 layer (network acces, internet, host to-host transport, dan application) yang masing-masing memiliki protokolnya sendiri-sendiri.

2.6.1 Fasilitas Internet

Internet memiliki banyak fasilitas, sebagaimana terlihat dibawah ini:

1. E-mail (*elektronik mail*)

Email adalah aplikasi internet yang memungkinkan para pengguna internet untuk saling berkirim pesan melalui alamat elektronik di internet. Penggunaan *email* memiliki sebuah kotak surat yang tersimpan dalam suatu *mailserver*. Dan *mailserver* atau kotak surat memiliki alamat sebagai pengenalan agar dapat berhubungan dengan *mailbox* lainnya

2. File Transfer

Fasilitas ini memungkinkan pengguna untuk melakukan pengiriman (*upload*) dan penyalinan (*download*) antara komputer local dengan komputer lain (*server*) yang terhubung dengan sebuah jaringan internet. Protocol standar yang digunakan disebut FTP (*File Transfer Protokol*). *Upload* adalah proses meletakkan file dari komputer kita ke komputer lain di internet. Sedangkan *Download* yaitu proses mengambil file dari komputer lain ke komputer kita. Untuk mempercepat download bisa menggunakan IDM.

3. IRC (Internet Relay Chat)

Atau Biasa disebut chat saja sekarang sudah ada dimana-mana. Banyak anak muda yang ketanggihan menggunakana ini karena dalam mengobrol kita bisa memilih dan memikirkan kata –kata yang mau di obrolkan dengan leluasa. Jika tidak mengerti dengan sebuah istilah, kita dapat langsung mencari di *Google*.

4. WWW (World Wide Web)

Dewasa ini *www* lebih di kenal dengan sebutan *web* saja. Karena merupakan aplikasi internet yang paling populer hingga banyak orang yang rancu dengan mengidentifikasikan pengertiannya dengan internet.

Web merupakan sebuah sistem dimana informasi berupa gambar, teks, suara dan lainnya tersimpan dalam sebuah internet *webserver* dipersentasikan dalam bentuk *hypertext*.

Dalam dunia internet selalu terdapat dua sisi yang saling mendukung , yaitu:

1. *server*, penyedia berbagai layanan termasuk *web*. Layanan *web* ditangani oleh sebuah aplikasi bernama *webserver*.
2. *client*, bertugas mengakses informasi yang disediakan oleh *server*. Pada layanan *web*, *client* berupa *web browser*

2.7 Website

Web adalah sebuah sistem dengan standar yang diterima secara universal untuk menyimpan, mengambil, memformat, dan menampilkan informasi melalui *client/server*. (Turban, Rainer, Potter 2003, p213)

Semua orang menggunakan *web* untuk segala sesuatu mulai dari pekerjaan kantor sampai dengan memesan makanan karena prosesnya yang cepat, murah, mudah dan menyenangkan. Tidak perlu meninggalkan meja. Bahkan orang yang suka belanja sekalipun mencari produk yang diinginkan di *web*.

Website hanyalah kumpulan file yang terletak pada sebuah komputer yang terhubung ke sebuah komputer dan komputer (yang kemudian disebut sebagai *server*) tersebut memberikan file yang ingin dilihat. Ini terdengar sederhana, tetapi umumnya komputer bisa tidak memiliki power dan software yang dibutuhkan untuk merespon semua permintaan yang dibutuhkan oleh pengunjung *website*. Dan walaupun

komputer/server tersebut sanggup, komputer akan membutuhkan koneksi/akses internet yang luar biasa cepat untuk melayani jumlah pengunjung website.

2.7.1 Fungsi Situs Website

Secara umum situs web memiliki fungsi sebagai berikut:

1. Fungsi Komunikasi.

situs web yang mempunyai fungsi komunikasi pada umumnya adalah situs web dinamis. karena dibuat menggunakan pemograman web (*server side*) maka di lengkapi fasilitas yang memberikan fungsi-fungsi komunikasi seperti *web mail, form contact, chatting, forum*.

2. Fungsi Informasi

Situs web yang memiliki fungsi informasi pada umumnya lebih menekankan pada kualitas bagian kontennya karena tujuan situs tersebut adalah menyampaikan isinya. Situs ini pada umumnya berisi teks dan grafik yang dapat di download dengan cepat.

3. Fungsi *Entertainment*.

Bila situs web berfungsi sebagai sarana hiburan maka penggunaan animasi gambar dan elemen bergerak dapat meningkatkan mutu presentasi desainnya, meski tetap harus mempertimbangkan kecepatan *downloadnya*. Beberapa fasilitas yang memberikan fungsi hiburan adalah *game online, film online, music online* dan sebagainya.

4. Fungsi Transaksi.

Situs web dapat dijadikan sarana transaksi bisnis, baik jasa maupun barang. Situs web ini menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik.

2.7.2 Jenis situs web.

Berikut ini beberapa jenis situs web yang dikelompokkan sesuai tujuannya:

1. Alat Pemasaran, Adalah sebuah situs yang dibuat dengan tujuan untuk mempromosikan dan memasarkan produk atau jasa layanan suatu perusahaan. Dapat juga berupa *company profile*. Pemasaran melalui internet lebih cepat sampai dan memiliki jangkauan yang jauh lebih luas.
2. Nilai Tambah. Biasanya sebuah halaman web pada awalnya disusun sebagai sarana promosi, karena media promosi di web lebih murah dibandingkan dengan media promosi konvensional seperti brosur, majalah atau koran. Sebuah situs web sering dibuat hanya sebagai adanya nilai tambah dalam perusahaan, mungkin perusahaan tersebut tidak terlalu membutuhkan sebuah situs tetapi menggunakannya hanya sekedar untuk mengikuti tren sehingga perusahaan tampak lebih modern.
3. Katalog. Sebuah situs digunakan sebagai catalog elektronik yang menampilkan produk-produk yang dijual perusahaan. Dengan menggunakan situs web, apalagi jika produk yang tersedia sangat banyak, peng-*update*-an dapat dilakukan dengan mudah, efisien dan lebih ekonomis. Beberapa hal yang harus diperhatikan dalam merancang situs web untuk katalog adalah informasi yang disampaikan jelas dan lengkap sehingga tujuan pemasarannya bisa tercapai.
4. E-Commerce. E-commerce merupakan suatu kumpulan yang dinamis antara teknologi , aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen dan komunitas tertentu melalui transaksi elektronik. Di sini perdagangan barang, pelayanan dan informasi dilakukan secara elektronik. Dalam membuat situs e-commerce harus dipahami hubungan antara brand dan costumernya. Situs harus bisa

memvisualkan aspek positif brand dan mengatur tempat dimana barang atau jasa tersebut untuk dijual.

5. E-Learning. E –Learning digunakan sebagai pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN dan internet) untuk menyampaikan isi pembelajaran, interaksi atau bimbingan. Ada juga yang menafsirkan e-learning sebagai bentuk pendidikan jarak jauh yang dilakukan melalui internet. Materi pembelajaran melalui media ini mengandung teks, grafik, animasi, simulasi, audio dan video.
6. Komunitas. Sebuah situs yang dibuat dengan tujuan untuk memungkinkan pengunjung berkomunikasi secara bersamaan . pengunjung bisa berbagi pengalaman, ide, cerita dan mencari teman baru atau untuk membuat suatu perkumpulan baru. Contohnya adalah www.facebook.com
7. Portal. Portal adalah aplikasi berbasis web yang menyediakan akses suatu titik tunggal dari informasi online terdistribusi, seperti dokumen yang di dapat melalui pencarian, kanal berita, dan link ke situs khusus. Untuk memudahkan penggunaanya biasanya disediakan fasilitas pencarian dan pengorganisasian informasi. Ada berbagai jenis portal yang mendukung beberapa komunitas dan berbagai macam servis. Portal memiliki berbagai macam komponen teknik dan infrastruktur untuk mendukung fungsinya.

2.7.3 Prinsip - Prinsip Desain Web.

Prinsip utama sebuah desain adalah kualitas atau karakteristik bawaan dalam berbagai bentuk seni, diantaranya adalah:

1. Keseimbangan. keseimbangan adalah hasil susunan satu atau lebih elemen desain sehingga anta yang satu dengan yang lainnya memiliki bobot yang sama. Keseimbangan adalah aturan dari kekontrasan visual. Bobot elemen biasanya dapat lebih dikenali dari ukuran objek dan kepadatan detail atau tekstur. Bila kekontrasan antar elemen terlalu besar, keserasian dan keseimbangan akan hilang. Hubungan antara proporsi dan ukuran harus benar-benar dipertimbangan pada saat ingin menghadirkan keseimbangan
2. Kontas. Kontras mudah dipahami, yaitu dengan melihat dari dua objek yang berlainan sehingga tampilan desain terkesan menonjol dan menarik perhatian. Pemberian kontras pada suatu objek haruslah kontras positif karena jika kontars yang diberikan adalah kontras negatif maka objek tersebut akan menjadi samar-samar bahkan tidak terlihat karena terserap oleh latar belakang
3. Konsistensi. Konsistensi membuat pengunjung merasa nyaman karena dapat menjelajah situs dengan mudah. Ketika pengunjung membuka suatu halaman situs yang bersifat konsisten, pengunjung akan langsung tahu kemana harus pergi dan pengunjung juga tahu sedang berada dimana. Konsistensi dapat diterapkan pada margin layout, huruf, warna dan terutama pada navigasi. Navigasi sebaiknya sama antara halaman satu dengan halaman lainnya. Konsistensi sangat efektif untuk membantu brand perusahaan.
4. Ruang kosong. Ruang kosong atau whitespace biasanya disebut dengan ruang negatif, suatu istilah yang menggambarkan suatu ruang terbuka diantara elemen-elemen desain. Ruang kosong memisahkan atau menyatakan elemen-elemen layout, menegaskan sebuah elemen atau sebagai tempat istirahat bagi mata. Ruang kosong bisa ditemukan diantara kata, paragraf huruf dan teks. Bisa juga diantara gambar dan

elemen disebuah halaman web. Ruang kosong sangat besar pengaruhnya dalam sebuah layout.

5. Warna. Warna dapat merebut perhatian, menarik, menolak bahkan juga bisa mempengaruhi emosi. Warna dapat menampilkan kesan pertama kepada pengunjung ketika menjelajah sebuah situs web karena warna adalah yang pertama kali dilihat terutama warna background. Warna dapat berfungsi dan berperan sangat banyak dalam sebuah desain web. Namun pada saat memilih warna kita perlu berhati-hati dengan efek-efek yang dipilih, karena dapat mengidentifikasi tema dari situs.
6. Tipografi. Tipografi adalah seni huruf , meliputi pemilihan huruf, penentuan ukuran yang tepat dimana teks dapat diputus, spasi jarak dan bagaimana teks dapat dengan mudah dibaca. Karena beragamnya bentuk huruf maka agak sulit untuk menentukan pengaruh apa yang bisa ditimbulkan pada pengunjung.

2.7.4 Kriteria Website Yang Baik.

1. *Usability*.

Usability adalah sebagai salah satu pengalaman pengguna dalam berinteraksi dengan aplikasi atau situs web pengguna dapat mengoperasikannya dengan mudah dan cepat. Situs web sebaiknya memenuhi lima syarat untuk mencapai tingkat usability yang ideal. Antara lain:

- a. Mudah untuk dipelajari. Letakkan isi yang paling penting pada bagian atas halaman agar pengunjung dapat menemukannya dengan mudah dan cepat.
- b. Efisien dalam penggunaan. Jangan menggunakan link terlalu banyak. Hantarkan informasi yang *user* butuhkan / berguna/ berharga dengan

- sedikit mungkin klik. Setiap link di website seharusnya tidak mengalihkan perhatian *user* dari pesan utama yang ingin disampaikan.
- c. Mudah untuk diingat. Situs sebaiknya jangan terlalu banyak melakukan perubahan terutama pada navigasinya.
 - d. Memiliki tingkat kesalahan yang rendah. Hindari link yang tidak berfungsi (*broken link*) atau halaman masih dalam proses pembuatan (*under construction*).
 - e. Kepuasan pengguna. Kepuasan adalah hal penting untuk diperhatikan untuk keberlangsungan situs web. Oleh karena itu sebuah website hendaknya enak dan nyaman untuk digunakan. *User* harus menemukan apa yang dicari dan dapat mendownloadnya dengan cepat. Mengetahui kapan saatnya selesai dan dapat dengan mudah memberitahukan site atau konten yang didapat kepada teman atau relasi.

2. *Sistem Navigasi.*

Kemudahan bernavigasi dalam situs web melibatkan sistem navigasi. Navigasi membantu pengunjung untuk menemukan jalan yang mudah ketika menjelajahi situs web. Memberitahu dimana pengunjung berada dan kemana pengunjung bisa pergi. Dengan demikian pengunjung dapat menemukan apa yang mereka cari dengan cepat dan mudah.

Navigasi dapat ditampilkan dalam berbagai media yaitu teks, image atau animasi. Navigasi dari image dapat menawarkan banyak sekali variasi misalnya dengan ikon, image, penggunaan huruf dan bentuk yang lebih bebas. Sedangkan navigasi dengan animasi memang sangat menarik karena berbagai interaktifitas visual yang ditawarkannya.

Syarat navigasi yang baik antara lain:

1. mudah dipelajari.
 2. tetap konsisten.
 3. memungkinkan *feedback*.
 4. muncul dalam konteks.
 5. menawarkan alternatif lain.
 6. memerlukan perhitungan waktu dan tindakan.
 7. menyediakan pesan visual yang jelas.
 8. menggunakan label yang jelas dan mudah dipahami.
 9. mendukung tujuan dan perilaku *user*.
3. Desain Visual

Kepuasan visual seorang *user* secara subjektif melibatkan bagaimana desainer situs web tersebut membawa mata *user* menikmati dan menjelajahi situs web melalui *layout*, warna dan bentuk dan tipografi. Grafis membuat halaman web menjadi indah tetapi juga dapat memperlambat akses dengan semakin besarnya ukuran *file*-nya.

Desain yang baik setidaknya memiliki komposisi warna yang baik dan konsisten, layout grafis yang konsisten, teks yang mudah dibaca, penggunaan grafis yang memperkuat isi teks, penggunaan animasi pada tempat yang tepat. Dan pada akhirnya secara keseluruhan membentuk suatu pola yang harmonis.

4. *Content*.

Sebaik apapun situs web, jika tidak disertai konten yang berguna dan bermanfaat maka kurang akan berarti. Konten yang baik, menarik, dan relevan akan berguna baik target audience. Gaya penulisan dan bahasa yang dipergunakan harus sesuai dengan web dan target audience. Konten

harus relevan dengan tujuan situs. Situs web sebaiknya tetao memiliki arsip dari konten dan data-data lama.

5. *Compability.*

Situs web harus kompatibel dengan berbagai perangkat tampilannya (browser). Beberapa syarat untuk meningkatkan kompatibilitas:

1. Tes di berbagai *browser*. *Browser* yang dipakai *user* bisa berbeda, versi yang berbeda atau dengan *setting* yang berbeda.
2. Pastikan situs web bekerja paling tidak di *platform* PC.
3. Jika memungkinkan, tawarkan pilihan tampilan situs, baik halaman web yang pelan (versi grafis atau animasi) atau cepat (versi HTML).
4. Kalau memakai *plug-in*, pastikan pengunjung dapat dengan mudah mendownloadnya.

5. *Loading Time.*

Sebuah penelitian yang dilakukan oleh zona research (april 1999) menyatakan bahwa 80% *user* akan menutup browser bila halaman web yang dibuka tidak tampil dalam 7-8 detik. Penelitian jupiter media metrix september 2001 – amerika serikat) mengatakan bahwa 40% *user* akan kembali mengunjungi situs yang tampil lebih cepat.

Sebuah situs web yang tampil lebih cepat kemungkinan besar akan kembali dikunjungi, apabila dengan konten dan tampilan yang menarik. Waktu download memang tidak hanya dipengaruhi desain tetapi juga koneksi, server, dll.

6. *Accesbility.*

Halaman web harus bisa dipakai setiap orang, baik anak-anak, orang tua, orang muda, termasuk orang cacat. Untuk hambatan fisik, dalam hal ini *user* dengan kekurangan indera penglihatan misalnya *user* yang penglihatannya kurang jelas memiliki kesulitan yang terletak pada ukuran teks sehingga kurang bisa membacanya maka dapat diatasi dengan memperbesar ukuran hurufnya. Pada orang-orang buta warna sebaiknya menghindari warna-warna rentan, bisa juga memberi warna yang kontras antara konten dan background.

7. *Interactivity.*

Interaktivitas adalah apa yang melibatkan pengguna situs web sebagai *user experience* dengan situs web itu sendiri. Dasar dari interaktifitas adalah *hyperlink (link)* dan mekanisme *feed back*. *Hyperlink* digunakan untuk membawa *user* kesumer berita, topic lebih lanjut, topik terkait atau lainnya. Seperti link yang berbunyi *more info about this, glosarry, related links* dan lainnya.

Sedangkan mekanisme *feed back*, contohnya adalh *critiques, comments, question, pooling / servey*. Keuntungannya, *user* bisa memberitahukan jika ada *missing link, dead link* atau kesalahan lain. *User* juga bisa memberi kritik atau saran agar web bisa lebih baik lagi.

2.8 Software Yang Digunakan

Berikut ini software yang mendukung dalam pengembangan teknologi web sebagai berikut: (Andi, 2002).

2.8.1 PHP

Aplikasi PHP, aplikasi PHP mutlak diperlukan karena system yang akan di buat menggunakan bahasa pemograman php.

Menurut dokumen resmi php, php singkatan dari *hypertext preprocessor*. Php merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. (Abdul Kadir 2009)

Php dirancang khusus untuk membentuk web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnay, anda bias menampilkan isi database. (Abdul Kadir 2009)

Kelahiran php bermula saat Rasmus Lerdof membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya yakni pada tahun 1994. Skrip-skrip ini yang menjadi cikal bakal php. Pada tahun 1995, Rasmus menciptakan php/f1 versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag html (*hypertext markup language*). Yang menarik, kode php juga bias berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks sambil jalan. (Abdul Kadir 2009).

2.8.1.1 Skrip PHP

Skrip php berkedudukan sebagai tag dalam bahasa html. Sebagaiman diketahui, html (*hypertext markup language*) adalah bahasa standar untuk membuat halaman-halaman web (Abdul Kadir 2009). Sebagai contoh, berikut adalah kode html:

```
<html>

<head>

<title> latihan pertama </title >

<head>

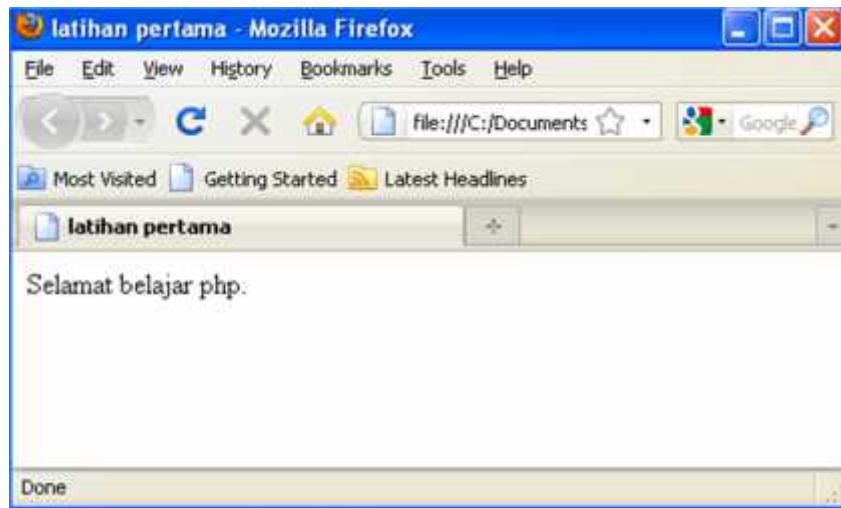
<body>

Selamat belajar php. <br>

</body>
```

</html>

Bila dijalankan melalui web browser, kode tersebut membentuk tampilan seperti berikut:



Gambar.2.29 Contoh hasil skrip PHP

2.8.1.2 Konsep Kerja PHP.

Model kerja html diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan url (uniform resource locator) atau dikenal dengan sebutan alamat internet, browser mendapatkan alamat dari web server, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh web server . (Abdul Kadir 2009)

Selanjutnya webbrowser akan mencari file yang dimana dan memberikan isinya ke web browser. Browser yang mendapatkan isinya segera melakukan proses penerjemahan kode html dan menampilkannya ke layar pemakai. (Abdul Kadir 2009)

2.8.2 MySQL

Mengenal MySQL MySQL adalah alah satu jenis database srver yang menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu, ia bersifat open source yang berada di berbagai platform.

MySQL termasuk jenis RDBMS (Relational Database Management system). Pada MySQL, sebuah database mengandung satu atau beberapa table yang juga terdiri dari sejumlah baris yang mengandung beberapa kolom.

2.8.2.1 Keunggulan MySQL (Rulianto Kurniawan 2010).

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta

dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.8.3 Apache

Apache adalah web server yang dapat dijalankan di banyak sistem operasi (Unix, Linux, Microsoft Windows dan platform lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. (Rulianto Kurniawan 2010)

Protocol yang digunakan untuk melayani fasilitas *web* atau *www* ini menggunakan HTTP. Apache memiliki fitur-fitur canggih seperti pesankesalahan yang dapat di konfigurasi, autentikasi berbasis data dan lain-lain. (Rulianto Kurniawan 2010).

